Enhancing Optimizer Stability: Momentum Adaptation of The NGN Step-size

Rustem Islamov[1] Niccolò Ajroldi[2] Antonio Orvieto[2,3,4] Aurelien Lucchi[1]

[1]University of Basel [2]Max Planck Institute for Intelligent Systems [3]ELLIS Institute Tübingen [4]Tübingen AI Center

Problem Formulation

We want to solve a finite-sum problem

$$\min_{x\in\mathbb{R}^d}\left\{ f(x) \overset{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n} f_i(x) \right\} \quad (1)$$

$f^* = \min_x f(x)$ non-convex; # of data points; $f_i^* = \min_x f_i(x)$; # model parameters; Empirical risk/loss; Loss associated with one data point; $\mathcal{S}$ is the set of global minimizers

- To solve (1), optimizer's hyperparameters should be tuned to deliver a good performance in practice.
- Tuning of optimizers for training large models is an expensive and a time-consuming process.

These issues motivate us to design algorithms that are robust to the choice of hyperparameters, such as the learning rate.

NGN Stepsize and Momentum

How to Add Coordinate-wise Adaptivity? For positive objective $f\colon \mathbb{R}^d \to \mathbb{R}$ we can approximate it around the point x as

$$f(x) = r^2(x) \;\Rightarrow\; f(x+\Delta) = r^2(x+\Delta) \approx (r(x) + \langle \nabla r(x), \Delta\rangle)^2.$$

This leads to NGN algorithm design

$$\begin{aligned} x^{k+1} &= \arg\min_{x\in\mathbb{R}^d} (r(x^k) + \langle \nabla r(x^k), x - x^k\rangle)^2 + \frac{1}{2\gamma}\|x - x^k\|^2 \\ &= x^k - \frac{\gamma}{1 + \frac{\gamma}{2f(x^k)}\|\nabla f(x^k)\|^2}\nabla f(x^k). \end{aligned}$$

NGN converges for any stepsize hyperparameter γ for convex and smooth objectives $\{f_i\}_{i\in[n]}$.

How to Add Momentum? We consider the two most popular versions of incorporating momentum to NGN algorithm

$$\text{Version 1}: \begin{cases} \gamma_k = \frac{\gamma}{1+\frac{\gamma}{2f_{S_k}(x^k)}\|\nabla f_{S_k}(x^k)\|^2} \\ m^k = \beta m^{k-1} + (1-\beta)\gamma_k \nabla f_{S_k}(x^k) \\ x^{k+1} = x^k - m^k \end{cases} \quad \text{(Heavy-ball)}$$

$$\text{Version 2}: \begin{cases} m^k = \beta m^{k-1} + (1-\beta)\nabla f_{S_k}(x^k) \\ \gamma_k = \frac{\gamma}{1+\frac{\gamma}{2f_{S_k}(x^k)}\|m^k\|^2} \\ x^{k+1} = x^k - \gamma_k m^k \end{cases} \quad \text{(Adam-like)}$$

We aim for a version that preserves the robustness of NGN and achieves accelerated convergence on quadratic functions.

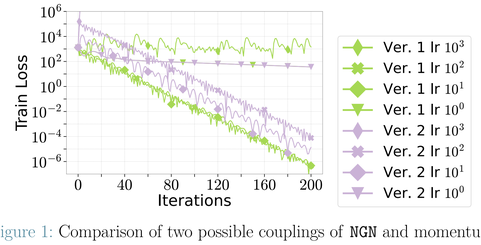

Figure 1: Comparison of two possible couplings of NGN and momentum.

Table 1:Summary of existing methods exploiting Polyak-type adaptive step-sizes and their convergence guarantees. **Mom.**=Supports momentum; **Diag.**=Supports diagonal step-sizes. $\sigma_{\text{int}}^2 := \mathbb{E}_S[f^* - f_S^*]$. $\mathcal{O}$ notation hides absolute, problem-dependent constants and logarithmic factors.

Method	Rate [(a)]	Mom.	Diag.	Comments
$\text{SPS}_{\max}$ [Loizou et al., 2021]	$\mathcal{O}(1/K + \sigma_{\text{int}}^2)$	✗	✗	Conv. to non-vanishing neighbourhood
ALR-SMAG [Wang et al., 2023]	$\mathcal{O}((1-\rho)^K + \sigma_{\text{int}}^2)$	✓	✗	Strong convexity Conv. to non-vanishing neighbourhood
Momo [Schaipp et al., 2024]	$\mathcal{O}(1/\sqrt{K})$	✓	✗	Bounded stoch. gradients Interpolation
Momo-Adam [Schaipp et al., 2024]	✗	✓	✓	Momo framework for Adam
$\text{MomSPS}_{\max}$ [Oikonomou and Loizou, 2024]	$\mathcal{O}(1/K + \sigma_{\text{int}}^2)$	✓	✗	Conv. to non-vanishing neighbourhood
NGN [Orvieto and Xiao, 2024]	$\mathcal{O}(1/\sqrt{K})$	✗	✗	–
IAM [Gower et al., 2025]	$\mathcal{O}(1/\sqrt{K})$	✓	✗	Knowledge of $f_i(x^*)$
NGN-M **[This work]**	$\mathcal{O}(1/\sqrt{K})$	✓	✗	–
NGN-MDv1 and NGN-MDv2 **[This work]**	✗	✓	✓	Combination of NGN-M and NGN-D
NGN-D **[This work]**	$\mathcal{O}(1/\sqrt{K})$	✗	✓	–

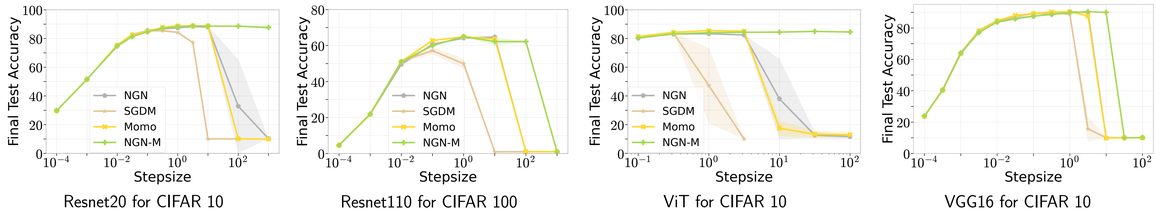

Figure 2:Comparison of NGN-M against several baselines with momentum when training small-scale models.

Algorithm 1: NGN-MDv1

Input: $x^0 = x^{-1} \in \mathbb{R}^d$, $\gamma > 0$, $\beta \in [0,1]$
For $t = 0, \dots, T-1$ **do**
  Sample $S_k \subseteq [n]$ and compute $f_{S_k}(x^k), \nabla f_{S_k}(x^k)$
  $v^k = \beta_2 v^{k-1} + (1-\beta_2)(\nabla f_{S_k}(x^k) \circ \nabla f_{S_k}(x^k))$
  $\mathbf{D}_k = \mathbf{I}$ (for NGN-M)
  $\mathbf{D}_k = \text{diag}\left(\varepsilon \mathbf{I} + \sqrt{v^k/(1-\beta_2^k)}\right)$ (for NGN-MDv1)
  $\gamma_k = \frac{\gamma}{1+\frac{\gamma}{2f_{S_k}(x^k)}\|\nabla f_{S_k}(x^k)\|^2_{\mathbf{D}_k^{-1}}}$
  $x^{k+1} = x^k - (1-\beta_1)\gamma_k \mathbf{D}_k^{-1}\nabla f_{S_k}(x^k) + \beta_1(x^k - x^{k-1})$

How to Add Coordinate Adaptivity to NGN?
We consider a similar model to approximate the objective around x^k with an arbitrary weight matrix $\mathbf{\Sigma}_k$

$$\begin{aligned} x^{k+1} &= \arg\min_{x\in\mathbb{R}^d} (r(x^k) + \langle \nabla r(x^k), x - x^k\rangle)^2 + \frac{1}{2\gamma}\|x - x^k\|^2_{\mathbf{\Sigma}_k} \\ &= x^k - \frac{\gamma}{1 + \frac{\gamma}{2f(x^k)}\|\nabla f(x^k)\|^2_{\mathbf{\Sigma}_k^{-1}}}\nabla f(x^k). \end{aligned}$$

One can choose $\mathbf{\Sigma}_k$
- by using gradient statistics, e.g., RMSprop preconditioner;
- Gauss-Newton approximation, e.g., AdaFisher matrix;

Theoretical Analysis of NGN-M

Convergence of NGN-M

Let f_i be positive, convex and L-smooth for all $i \in [n]$, and the errors $\sigma_{\text{int}}^2 := \mathbb{E}_S[f^* - f_S^*]$ $\sigma_{\text{pos}}^2 := \mathbb{E}_S[f_S^*]$ be bounded. Then we have

$$\mathbb{E}[f(\overline{x}^K) - f^*] \le \frac{R(1+2\gamma L)^2}{\gamma K} + 8\gamma L(1+2\gamma L)^2\sigma_{\text{int}}^2 + 2\gamma L \max\{0, 2\gamma L - 1\}\sigma_{\text{pos}}^2,$$

where $R := \|x^0 - x^*\|^2$ and $\overline{x}^K \sim \text{Unif}[\{x^0, \dots, x^{K-1}\}]$, and $\beta = \frac{\lambda}{1+\lambda}$ with $\lambda \le \min\{\gamma L, 0.5(1+\gamma L)^{-1}(1+2\gamma L)^{-1}\}$.

- If we set $\gamma = \mathcal{O}(1/\sqrt{K})$, we obtain the convergence rate of order $\mathcal{O}(1/\sqrt{K})$.
- No restriction on the LR hyperparameter γ as long as the momentum parameter is small enough.
- In contrast to prior work, the convergence is obtained without strong assumptions such as interpolation condition $\sigma_{\text{int}}^2 = 0$.
- Small-β requirement can be moved to the LR hyperparameter γ under interpolation condition.
- Decaying LR hyperparameter γ as $\mathcal{O}(1/\sqrt{k})$ removes the need for knowing the total number of iterations K.

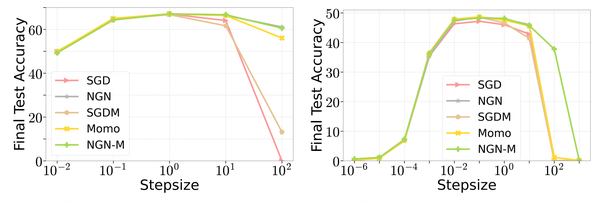

Figure 3:Comparison of NGN-M against several baselines with momentum when training Resnet18 on ImageNet datasets.

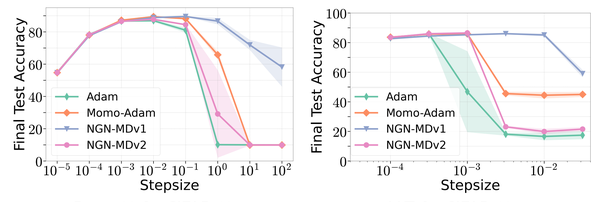

Figure 4:Comparison of NGN-MDv1 against several baselines with momentum and preconditioning when training Resnet18 on ImageNet datasets.

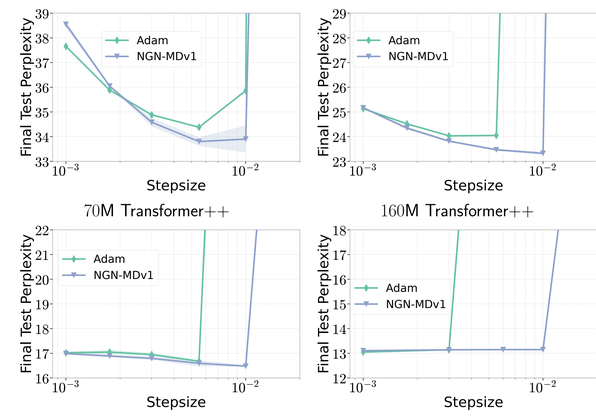

Figure 5:Comparison of stability to the LR hyperparameter across model sizes and optimizers in language modeling.

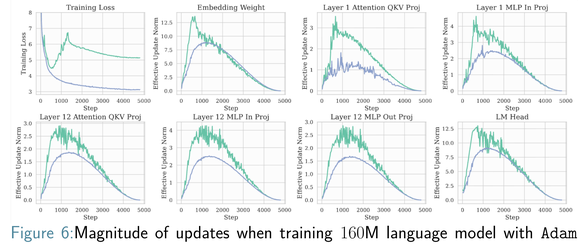

Figure 6:Magnitude of updates when training 160M language model with Adam and NGN-MDv1 and step-size hyperparameter 0.01.

Table 2:Train time of AdamW and NGN-MDv1 when training language models.

Model	Method	Time per Iteration (sec)	Time per Optimizer Update (sec)
70M	AdamW	$1.63_{\pm 0.01}$	$0.0048_{\pm 0.0002}$
	NGN-MDv1	$1.65_{\pm 0.01}$	$0.0130_{\pm 0.0002}$
160M	AdamW	$3.33_{\pm 0.03}$	$0.0088_{\pm 0.0003}$
	NGN-MDv1	$3.37_{\pm 0.02}$	$0.0239_{\pm 0.0003}$
410M	AdamW	$8.41_{\pm 0.06}$	$0.0838_{\pm 0.0009}$
	NGN-MDv1	$8.68_{\pm 0.06}$	$0.2154_{\pm 0.0007}$